MICRO CONTROLLER BASED ANTENNA CONTROL

By

M RAMESHA

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JUNE 1995

MICRO CONTROLLER BASED ANTENNA CONTR

A Thesis submitted in partial
fulfilment of the Requirements
for the Degree of

DIIT

by

M Ramesha

DEPARTMENT OF ELECTRICAL ENGINEERIN(
INDIAN INSTITUTE OF TECHNOLOGY KANPUI

June, 1995

CERTIFICATE

It s rtifi d th t th work c ntain d in th thes s titled Mi ro c ntroller Bas d Antenna Cntr l by M Ram sha (Registiation number 9412403) has b en ca ri d ut under my up vision and that this work h not b en submitt d ls whe e for a degree

30/5/95

Dr Sub r R y

Assistant Proffe s r

Department f El t ical Engin ering

IIT Kanpur

ABSTRACT

A Microcontroller based satellite tracking system is discussed. Interfacing Microcontroller to the IBM PC is explained. Software and hardware for the movement of stepping motor is presented. A driver system to amplify power of control signals is discussed.

ACKNOWLEDGEMENTS

I am pertı ularly grat ful to my guıd Dr Subır R y who nspıt of my hort mıngs e tended hıs kınd supp rt and ınvaluable dvıc I am greately ınd bted to Srı A C J shı fo hıs advıce guıd nc and op ratıon at very step th ought the d velopm nt f thıs the ıs

I am greateful t my f ends at SBRA f r theı co op ratıon and b hınd the sc en parts played I sınc rely thank Vın d Kamat for renderıng hı h lpıng hand wıthout faıl when v r I s ught ıt I thank M s P abha fo h h lp to typ thesıs I am greateful t D d rshan for provıdıng x llent opp rtunıty

I shuld th nk my wıf for he love and supp rt durıng dıffıcult tıme

Dat

ONTENTS

L I S T   O F   F I G U R E S

CHAPTER 1

INTRODUCTION

Motivation to develop to study and to extend system was the open loop satellite antenna tracking system in Doordarshan This system can track satellite by changing azimuth and elevation angle manually step by step Antenna position is optimized by observing the received signal quality

Increased usage of satellite communication for TV and telephony has created a need to go for improvements in satellite tracking systems Today Micro controllers have become cheaper, efficient and faster The advent of VLSI technology has made all this possible The use of Micro controller for satellite tracking can avoid manual errors, and faster tracking

The developmental set up consists of an IBM compatible PC, Micro controller, Driver circuitry and Stepper Motors For interfacing the PC and Micro controller serial communication interface RS 232C, has been used Software for down loading assembly level program from PC to Micro controller is based on the S-record S19 files

ORGANIZATION OF THESIS

The present thesis is organized in the following five chapters, starting with the current one on introduction Chapter 2 reviews the Micro controller system and its built in analog to digital converter Chapter 3 contains an introduction to Stepper

Motor Chapter 4 is devoted to the proposed Micro controller based satellite tracking system This includes algorithms, hardware requirements, driver circuitry and implementation details Finally chapter 5 concludes with some suggestions for future work

CHAPTER 2

INTRODUCTION TO MICROCONTROLLER

Central processing unit (CPU) which is responsible for executing all software instructions in their programmed sequence The architecture of the M68HC11 CPU uses memory mapped I/O Figure (2 1) shows the block diagram and Figure (2 2) shows the programming model of the CPU It contains six registers These are

(1)Two accumulators A and B

(2)Two index registers X and Y

(3)One stack pointer SP

(4)One program counter PC

Details of these registers, and the condition code register can be obtained from the manual

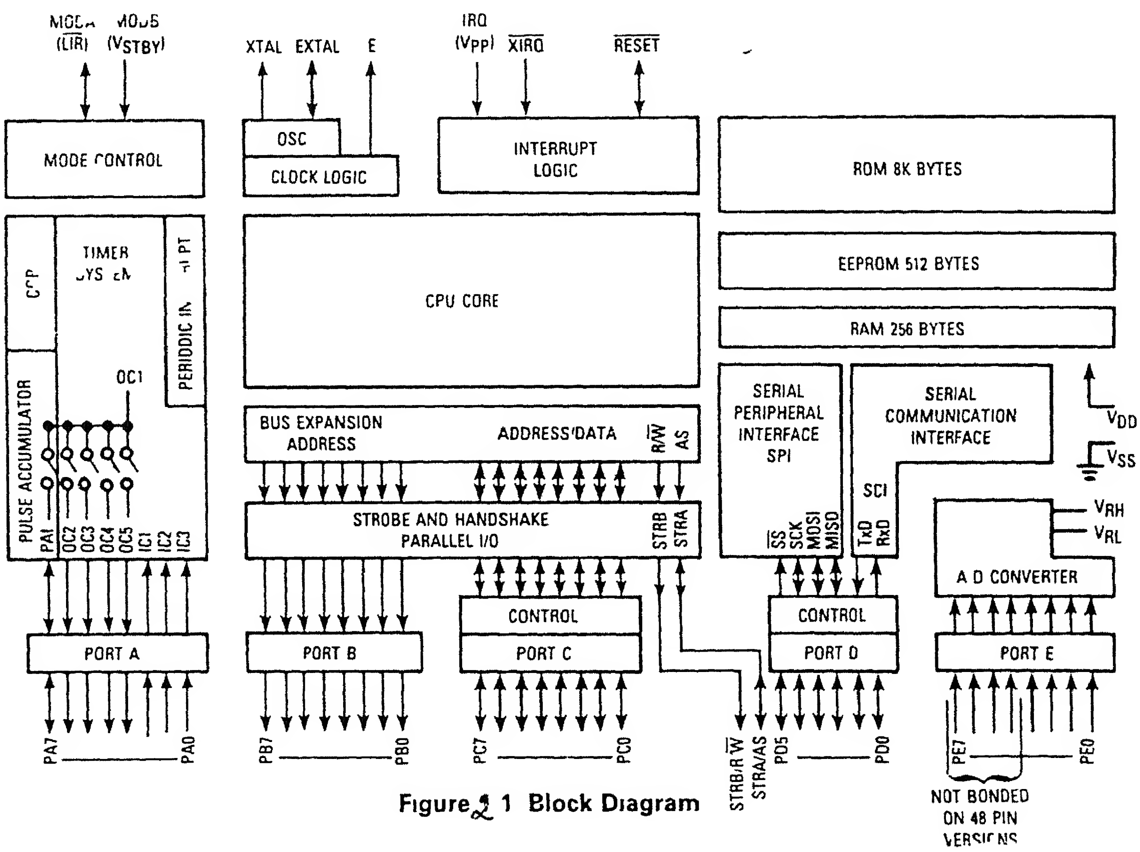


Figure 2 1 Block Diagram

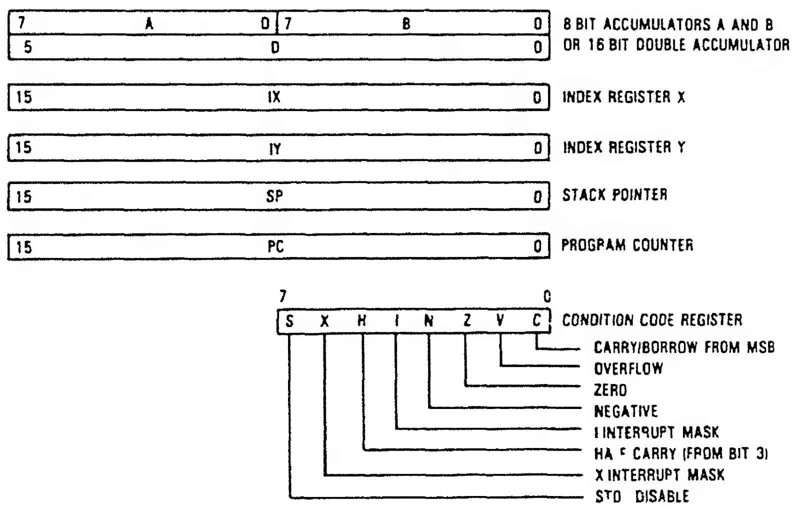


Fig 2 2 Programing Model

2 1 ADDRESSING MODES

The six addressing modes used to reference memory are as follows

2 1 1 Immediate addressing

In the immediate addressing mode, the actual argument is contained in the byte(s) immediately following the instruction in which the number of bytes matches the size of the register

2 1 2 Direct addressing

In the direct addressing mode, the least significant bit

of the effective address of the instruction appears in the byte following the op code The high order byte of the effective address is assumed to be $00 and is not included as an instruction byte, to save program memory space and execution time Therefore direct addressing mode is limited to $0000 - $00FF area of memory space

2 1 3 Extended addressing

In the extended addressing mode the effective address of the instruction appears explicitly in the two bytes following the op code

2 1 4 Indexed addressing

In the indexed addressing mode, either index register X or Y is used in calculating the effective address In this case, effective address of the data variable depends on the current content of index register X or Y, and a fixed 8 bit unsigned offset contained in the instruction That is when offset is added to the current value in the index register, yields effective address of the operand If no offset is specified the machine code will contain $00 in the offset byte

2 1 5 Inherent addressing

In the inherent addressing mode every thing needed to execute the instruction is inherently known by the CPU The operands are present in CPU registers and thus are not fetched from memory These instructions are usually one or two bytes

Example	Operation	Comments
	ABA	A + B ⇒ A

2 1 6 Relative addressing

The relative addressing is used only for branch instructions If the branch condition is true, the contents of the 8 bit signed byte following the op code are added to the contents of the program counter to form an effective branch address, otherwise control proceeds to the instruction, immediately following the branch instruction

The limited range of branches (-128 to +127 locations) is more than adequate for most situations In cases where this range is too short, a jump instruction must be used For every branch, there is a branch for the opposite condition

2 2 General Description of the MC68HC11 A8

This Micro controller operates with nominal bus speed of 2 MHz Major peripheral functions are provided on-chip An eight channel analog to digital converter (A/D) is included with eight bits of resolution, and a synchronous serial communications interface (SCI) and a separate synchronous serial peripheral interface (SPI) are included The main 16 bit free running timer system has three input capture lines, five output compare lines and a real-time interrupt function An 8 bit pulse accumulator subsystem can count external events or measure external periods

Self monitoring circuitry is included on chip to protect against system errors A computer operating property (COP) watchdog system protects against software failures A clock monitor system generates a system reset in case the clock is lost or runs too slow An illegal op code interrupt is generated if an illegal op code is detected

Two software controlled power saving modes WAIT and STOP are available to conserve additional power These modes make the M68HC11 family especially attractive for auto-motive and battery driven applications

The MC68HC11A8 Micro controller includes an 8 channel multiplexed input, successive approximation, analog to digital converter Two dedicated lines (VR_L, VR_H) are provided for the reference voltage inputs This ensures full accuracy of the A/D conversion The 8-bit A/D converter has an error of ± 1LSB all error sources included,and accepts analog inputs which range from VR_L to VR_H Each conversion is accomplished in 32 MCU E clock cycles That is $1 / (2 * 10^6 * 32)$ sec time

An input voltage equal to VR_L converts to $00 and an input voltage equal to VR_H converts to $FF, with no overflow indication Multiplexer allows the single A/D converter to select one of the four channels correspond to port E input lines to the MCU

2 3 ANALOG TO DIGITAL CONVERTER

The following illustration show the OPTION control register for reference

OPTION	ADPU	CSEL	IRQE	DLY	CME	___	CRI	CRO
$1039								

2 3 1 A/D Charge Pump

A charge pump on the chip develops about 7 or 8 Volts

and this high voltage is used to drive the gates of the analog switches in the input multiplexer and capacitor array

The A/D charge pump is disabled after coming out of reset later on is turned on by setting the A/D power up (ADPU) control bit in the OPTION control register Before using the A/D system

2 3 2 MC68HC11A8 A/D System Control Logic

The A/D system on the MC68HC11A8 consists of a successive approximation A / D converter, an input multiplexer to select one of 16 channels, and sophisticated control circuitry to configure and control conversion activities Four separate result registers are included with control logic that implements automatic conversion or the analog value in a selected channel four times and the result will be stored in four A / D result registers Alternatively if the multi channel option is selected where in there are four different inputs the analog values are converted sequentially and stored in the respective A / D result registers Conversion sequences are configured to repeat continuously or to stop after one set of four conversions A / D result registers should not normally be used before the Conversion Complete Flag (CCF) is set at the end of the fourth conversion

$1030	CCF	O	SCAN	MULT	CD	CC	CB	CA	ADCTL

2 3 3 Conversions Complete Flag (CCF)

This read only status indicator is set when all four A/D result registers contain valid conversion results. Each time the ADCTL register is written, this bit is automatically cleared to zero and a conversion sequence is started. Bit 6 is not implemented.

2 3 4 Continuous Scan Control (SCAN)

When this control bit is cleared, the four requested conversions are performed at once to fill the four result registers. When this control bit is set, conversions continue in a round robin fashion with the result registers being updated as data becomes available.

2 3 5 Multiple or Single Channel Control (MALT)

When this bit is cleared, the A / D system is configured to perform four consecutive conversions on the single channel specified by the four channel select bits CD through CA.

When this bit is set, the A / D system is configured to perform a conversion on each of four channels where each channel result register corresponds to one channel.

ANALOG TO DIGITAL CHANNEL ASSIGNMENTS

CD	CC	CB	CA	Channel signal
0	0	0	0	PE_0
0	0	0	1	PE_1
0	0	1	0	PE_2
0	0	1	1	PE_3

2 3 6 A/D Result Registers ADR1, ADR2, ADR3, ADR4

The A / D result registers are read only registers used to hold an 8 bit conversion result Writes to these registers have no effect Data in the A / D result registers is valid when the CCF flag bit in ADCTL register is set, indicating a conversion sequence is complete

2 4 Analog To Digital Conversion System

The MC68HC11A8 analog to digital converter system uses an all capacitive charge redistribution technique for conversions

Fig (2 3) shows a simplified circuit to perform a 4 bit successive approximation A/Dconversion using charge redistribution The actual circuit used in the MC68HC11A8 includes several additions to this simple circuit to improve quality and simplify manufacturing

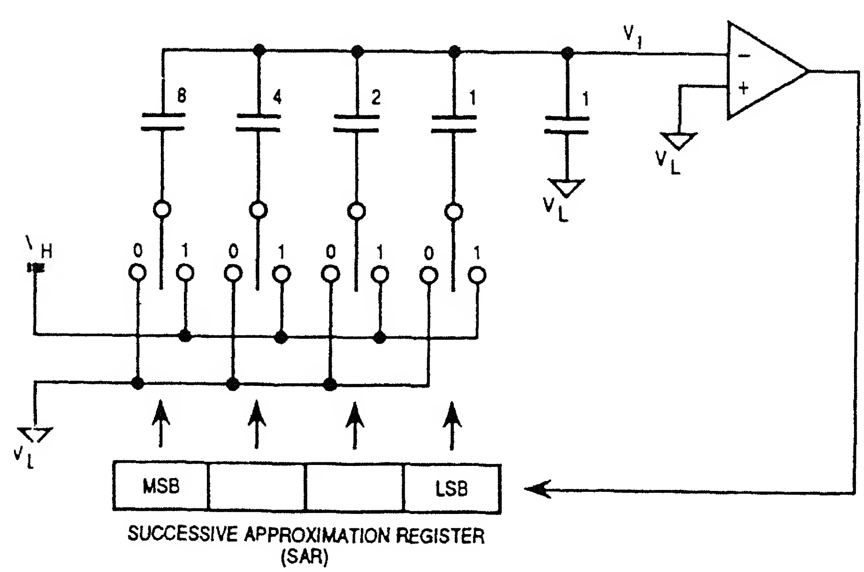
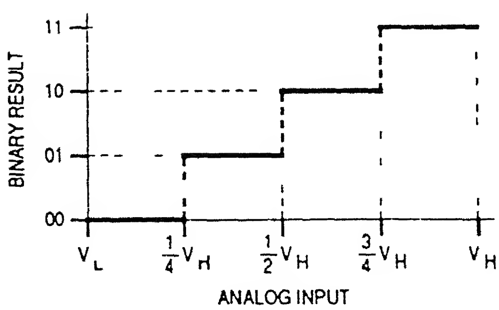


Fig 2 3 A/D Converter Circuit

Since the capacitive charge redistribution technique depends upon capacitive ratios rather than absolute capacitive values, the capacitors in fig (2 3)are marked in units

Transfer characteristics of the above circuit used for A/D conversion is as follows,



Fig(2 4) A/D Transfer Characteristics

An analog input of 1/4 VH produces a digital result of 01, but an analog input of 1/8 VH yields a digital result of 00, which is in error by 1/8 VH or 1/2 LSB This quantisation error is an un avoidable consequence of any A/D converter

Because of this in-accuracy, motor will be rotated only when the difference between successive samples is more than 2LSB Because absolute accuracy is ±1LSB, diference between the actual input voltage and the full-scale weighted equivalent of the binary out put code All error sources included Following is the table of A/D converter characteristics

(V_{DD} = 5 0 Vdc ± 10% V_{SS} − 0 Vdc $T_A = T_L$ to T_H 750 kHz E 2 1 MHz unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by the A/D	8	—	—	Bits
Non Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	1 2	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual A/D for Zero Input Voltage	-	—	1/2	LSB
Full Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full Scale Input Voltage	—	—	± 1/2	LSB
Total Unadjusted Error	Maximum Sum of Non Linearity Zero Error and Full Scale Error	—	—	+ 1 2	LSB
Quantization Error	Uncertainty Due to Converter Resolution	—	—	+ 1/2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full Scale Weighted Equivalent of the Binary Output Code All Error Sources Included	—	—	± 1	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (see Note 2)	V_{RL}	—	V_{DD} + 0 1	V
V_{RL}	Minimum Analog Reference Voltage (see Note 2)	V_{SS} − 0 1	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (see Note 2)	3	—	—	V
Conversion Time	Total Time to Perform a Single Analog to Digital Conversion a E Clock b Internal RC Oscillator	 — —	 32 —	 — t_{cyc} + 32	 t_{cyc} μs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed		
Zero Input Reading	Conversion Result when $V_{in} = V_{RL}$	00	—	—	Hex
Full Scale Reading	Conversion Result when $V_{in} = V_{RH}$	—	—	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time a E Clock b Internal RC Oscillator	 — —	 12 —	 — 12	 cyc μs
Sample/Hold Capacitance	Input Capacitance during Sample PE0 PE7	—	20 (Typ)	—	pF
Input Leakage	Input Leakage on A/D Pins PE0 PE7 V_{RL} V_{RH}	— —	— —	400 1 0	nA μA

NOTES

1 Source impedances greater than 10 KΩ will adversely affect accuracy due mainly to input leakage

2 Performance verified down to 2 5 V ΔV_R but accuracy is tested and guaranteed at ΔV_R = 5 V ± 10 /

Table 2 1 A/D Converter Characteristics

CHAPTER 3

INTRODUCTION TO STEPPER MOTOR

The principle of a stepping motor can be explained by considering a three-phase stepping motor Fig(3 1)shows the cross sectional structure of a single stack variable-reluctance three phase stepping motor

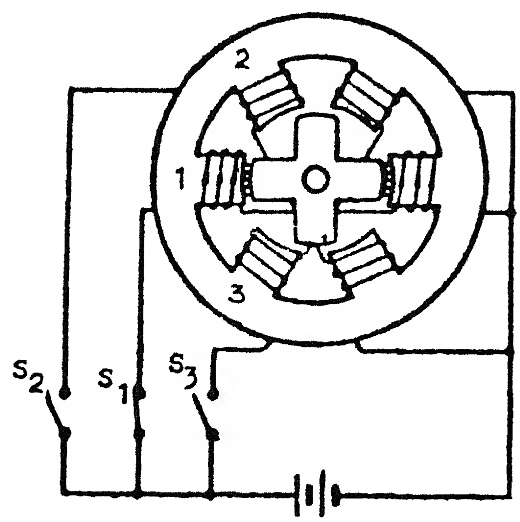


Fig 3 1 Cross Sectional View of Stepper Motor

The stator core has six salient poles, or teeth while the rotor has four poles The cores are of soft steel Three sets of windings are arranged as shown in fig(3 1) Each set has two coils connected in series A set of windings is known as "phase" and consequently this machine is a three phase motor Current is supplied from a DC power source to the windings via switches

S_1,S_2,and S_3 In fig(3 1)the winding of phase 1 is supplied with current through S_1, in other words phase 1is excited A magnetic flux occurs in the air gap due to the excitation of phase 1 The two stator salient poles of phase 1 being excited, are in alignment with, two of the four rotor teeth This state is referred to as an equilibrium state When S_2 is also closed, the phase 2 is also, excited in addition to phase 1, magnetic flux builds up at the stator poles of phase 2 and a anti-clock wise torque is created owing to *tension* in the inclined magnetic field lines as shown in fig(3 2)

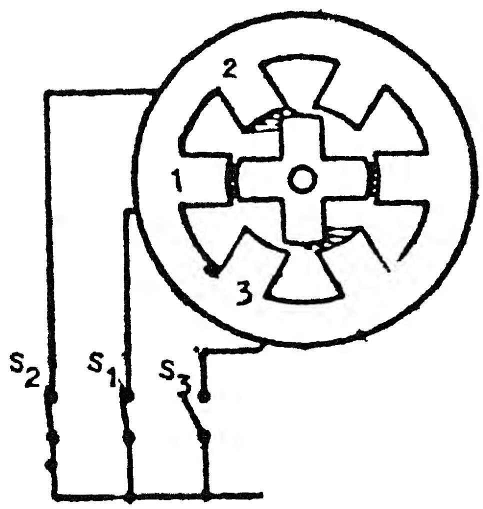


Fig 3 2 Magnetic Field Lines After Excitation

The rotator will rotate through a fixed angle and reaches next state in fig(3 3)

This angle is termed as *step angle* Step angle 15° corresponds to one switching operation in the present case

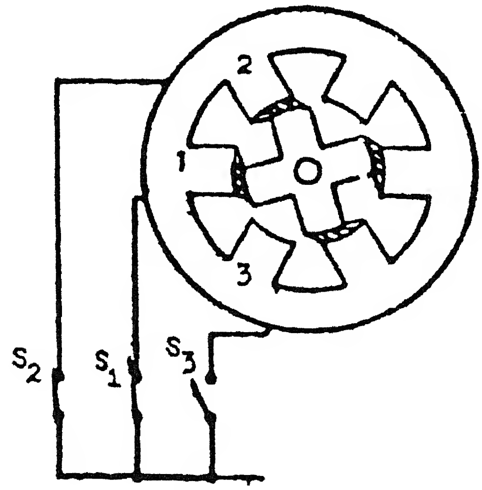


Fig 3 3 **Magnetic Field After One Step Rotation**

If S1 is now opened to de energize phase 1, the rotor will travel another 15° to reach the state shown in fig(3 4) Now it has come to the state of equilibrium

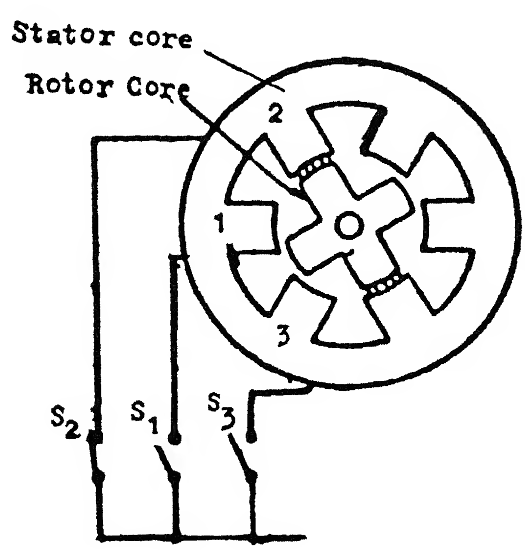


Fig 3 4 Magnetic Field in Equilibrium State

Thus the angular position of the rotor can be controlled in units of the step angle by a switching process If the switching is carried out in sequence, the rotor will rotate with a stepped motion The average speed of the rotor can also be controlled by the switching process We have used a micro controller system based on MC68HC11A8 To generate the switching signals, and fabricated the driving circuit for driving the motor

The stepping motor used in our system is a four-phase stepping motor It has 200 steps The cross sectional view of a four-phase motor is shown in fig(3 5)

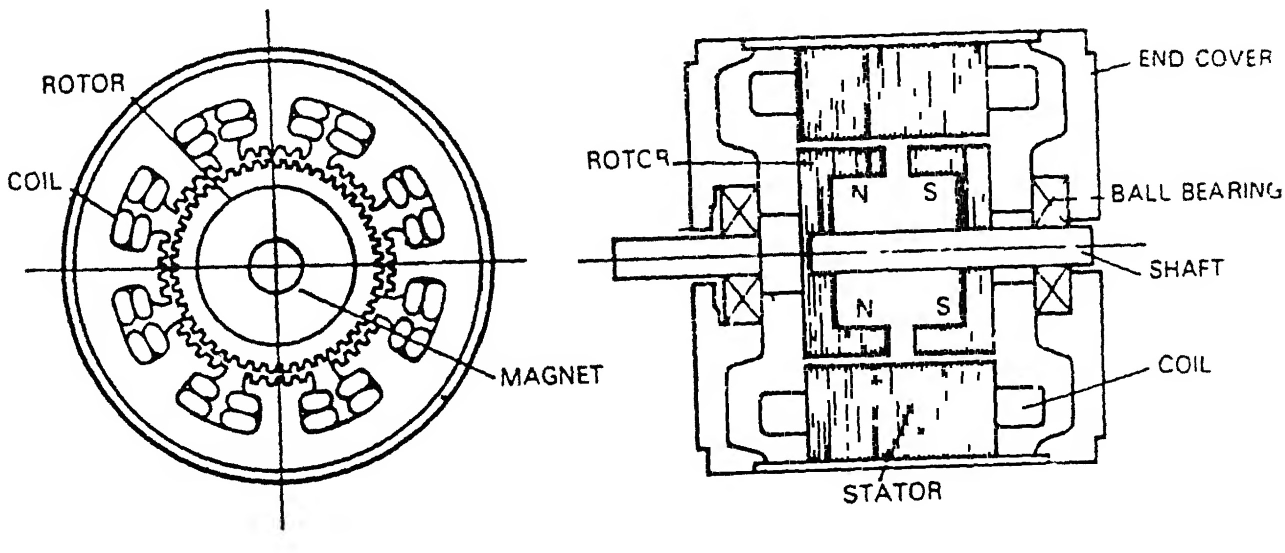


Fig 3.5 Four-phase Motor Cross Sectional View

3 1 SALIENT FEATURES OF STEPPER MOTORS

3 1 1 Small Step Angle

As discussed before a stepping motor rotates through a

fixed angle, step angle, for each pulse The step angle is expressed in degrees The smaller the step angle, the higher the resolution of position The number of steps per revolution is termed as *step number*

The step angle θ_s is related to the step number S by

$$S=360/\theta_s=mN_r$$

Where *m* is the number of phases N_r number of rotator teeth

A standard four phase motor has a step number of 200, (some precision motors are designed attain one revolution with 500 or 1000 steps However, the step angle in some motors are as large 90°,45° or 15°)

3 1 2 High Positioning Accuracy

Stepping motors are designed so that they rotate through a pre-determined step angle in response to a pulse signal and come to rest at a precise position The quality of a stepping motor depends on the accuracy in positioning which in turn depends on the machine characteristics, rotor and stator accuracy

3 1 3 High Torque To Inertia Ratio

It is desirable that a stepping motor moves as fast as possible in response to an in put pulse train Not only a quick start but a quick stop is required for a stepping motor If the pulse train is interrupted while the motor is running at a uniform speed, the motor should be capable of stopping at the position specified by the last pulse It is large torque to rotor inertia ratio in stepping motors, makes them advantageous to conventional

electrical motors

3 1 4 Stepping Rate And Pulse Frequency

The speed of a stepping motor is characterized by *stepping rate* and is defined as the number of steps per second It is expressed in hertz since in most stepping motors, the number of pulses applied to the logic sequence equals the number of steps, the speed may be expressed in terms of pulse frequency

Stepping rate does not specify the absolute speed the conventional electrical machines are characterized by rotational speed, expressed in terms of revolution per minute The relation between the rotational speed and stepping rate is given by

$$n = 60f/S$$

Where S= Step number

f= Stepping rate

n= Rotational speed

CHAPTER 4

INTERFACING

4 1 Interface Cabling

The micro controller board (EVM) is required to work in association with a PC system Commands are issued to the micro-controller using this PC The interfacing of the micro controller with PC system is therefore essential

The following hardware items are required for interfacing micro controller with the IBM-PC system

IBM-PC(with RS-232 C serial port)

Fabricated RS-232 C cable assembly

The IBM-PC must have an RS-232 C serial port to interface with EVM The IBM-PC must have at least one disk drive, with a disk operating system (DOS) version 2 0 or higher

Fig 4 1 illustrates the fabricated RS-232 C cable assembly this cable must be connected between the IBM-PC COM1 or COM2 I/O port connector and the EVM terminal I/O port connector

4 2 IBM-PC HARDWARE INTERFACING

To interface the IBM-PC to the EVM following steps have to be performed

1 Connect fabricated RS-232 C cable assembly as shown in Fig 4 1 from the IBM-PC COM1 or COM2 I/O port connector to the EVM terminal I/O port connector

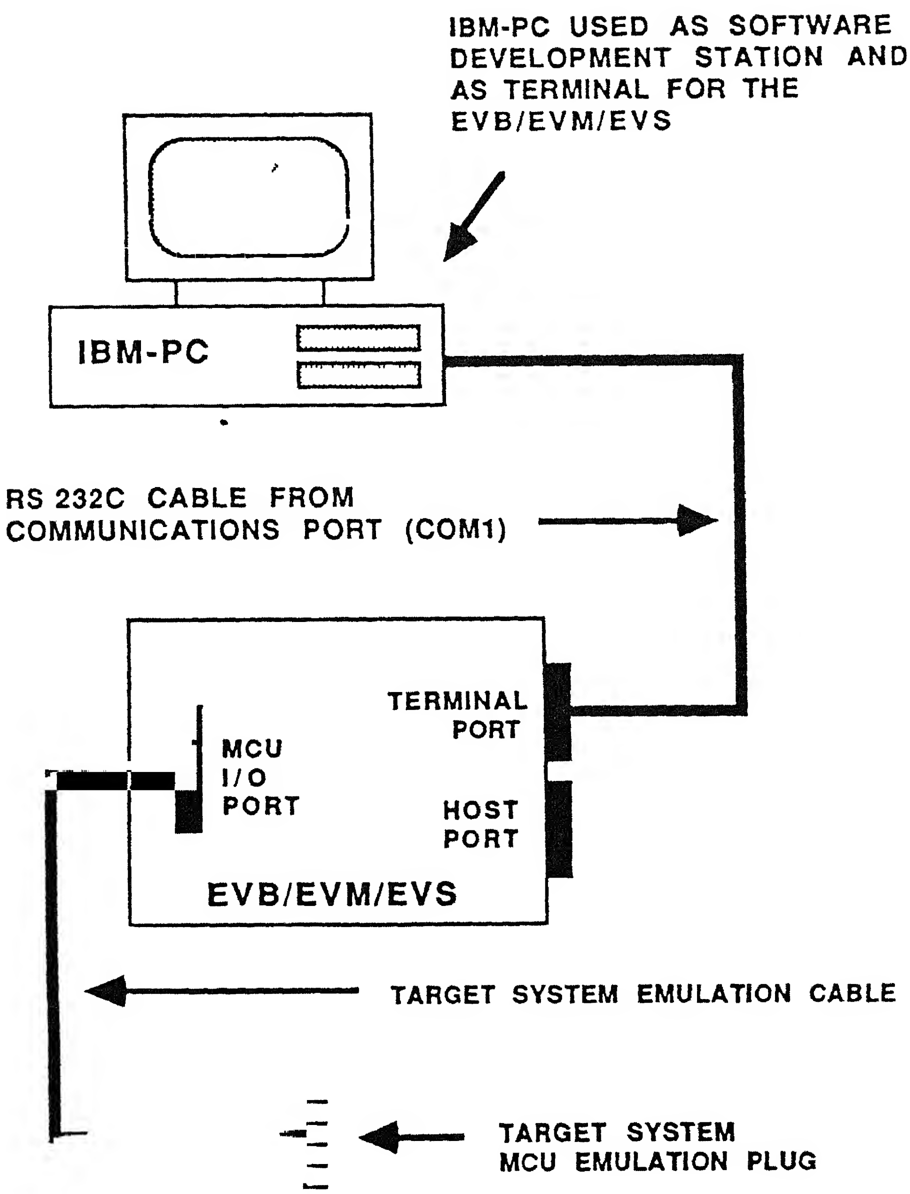


Fig 4 1 Interface of EVM with IBM-PC

Fig 4 2 illustrates the fabricated RS-232C cable assembly This cable assembly connected between the IBM-PC COM1 I/O port connector and the EVM terminal I/O port connector

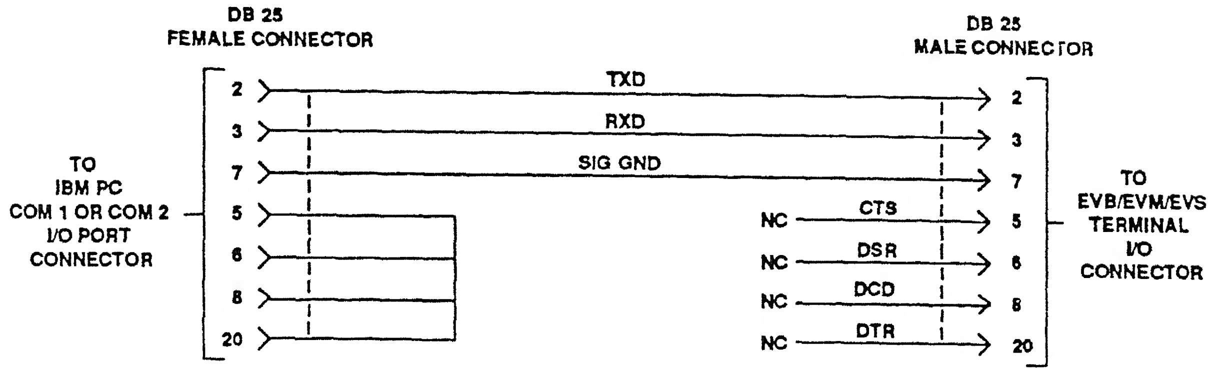


Fig 4 2 RS-232C Cable Assembly

2 For M68HC11 EVM, on jumper header J14,jumper pins 1 and 2 for semi-hand shake operation

3 Connect power supply to EVM

4 3 IBM-PC SOFTWARE REQUIREMENTS

To communicate with the EVM via the terminal port, the IBM-PC must appear as a port to the EVM The down loading program(Terminal emulator software) used for IBM-PC is PROCOMM

4 4 IBM-PC S-RECORD FILE GENERATION

The S-Record format for output modules was devised for the purpose of encoding programs in a printable format for transportation between computer systems The transportation process of the system can thus be visually monitored, on the monitor, S-record file moves vertically from bottom to top and the S-Records can be more easily edited

4 5 S-RECORD CONTENT

S-Records are essentially character stings made of several fields which identify the S-Record type, record length, memory address, code/data, and checksum Each type of binary data is encoded as a two character hexadecimal number

Eight types of S-Records have been identified to accommodate the several needs of the encoding, transportation, and decoding functions

The EVM monitor supports only the S1 and S9 records All data before the first S1 record is ignored There after all records must be S1 type until the S9 record terminates data transfer

S-Record format module for EVM contains S-Records of the following type

S0 The header record for each block of S-Records The address field is normally zeros

S1 A record containing code/data and the two byte address at which the code data is to reside

S9 A termination record for a block of S1 records The address

field may optionally contain 2-byte address of the instruction to which control is to be passed If not specified, the first entry point specification encountered in the object module input will be used There is no code/data field

Several programs are available for down loading a file in S-record format from a host type to an 8-bit or 16-bit microprocessor based system

Shown below is a typical S-record format module,

S00600004844521B

S1130000285F245F2212226A000424290008237C2A

S11300100002000800082629001853812341001813

S113002041E900084E42234300182342000824A952

S107003000144ED492

S9030000FC

The above module consists of S0 header record, four S1 code and data records and an S9 termination record

The S0 header record is comprised of the following character pairs

S0 S-record type S0,indicating a header record

06 Hexadecimal 06, indicating six character pairs (or ASCII bytes)follow

00 Four character two byte address field, zeros

00

48

44 ASCII H, D,and R-"HDR"

52

1B Checksum of S0 record

The first S1 code/data record is explained as follows

S1 S-record type S1,indicating a code/data record to be loaded/verified at a 2-byte address

13 Hexadecimal 13 (decimal 19) indicating 19 character pairs, representing 19 bytes of binary data, follow

00 Four-character 2-byte address field, hexadecimal address 0000, indicates location where the following data is to be loaded

The next 16 character pairs are the ASCII bytes of the actual program code/data In this assembly language example, the hexadecimal op codes of the program are written in sequence in the code/data fields of the S1 records

OP CODE	INSTRUCTION	
28 5F	BHCC	$0161
24 5F	BCC	$0163
22 12	BHI	$0118
22 6A	BHI	$0172
00 04 24	BRSET	0,$04,$012F
29 00	BHCS	$010D
08 23 7C	BRSET	4,$23,$018C

(Balance of this code is continued in the code/data fields of the remaining S1 records, and stored in memory location 0010,etc)

2A Checksum of the first S1 record

The second and third S1 code or data records each also contain

$13(19) character pairs and are ended with check sums 13 and 52,respectively The fourth S1 code/data record contains 07 character pairs and has a check sum of 92

The S9 termination record is explained as follows

S9 S-record type S9,indicating a termination record

03 Hexadecimal 03,indicating three character pairs (3 bytes) follow

00 Four-character 2-byte address field, zeros

00

FC Checksum of S9 record

4 6 S-RECORD FILE GENERATION

To create S-record file perform the following steps

1 Create a text file with the IBM-PC text editor

2 Type the above S-record format module

3 Name the S-record file "file name S19" in present directory used

4 7 DOWN LOADING S-RECORD FILE

For performing an IBM-PC to EVM down, loading operation, we use PROCOMM program that enables the host computer system to emulate a terminal

PROCOMM is a screen oriented communication program which functions as a dumb terminal For down loading Motorola S-records from an IBM-PC to the EVM, this function is all that is required

4 8 CONNECTOR SIGNAL DESCRIPTION

The EVM provides two input/out put connectors J_1 and J_3, that are used to inter connect the EVM to a target system As the Micro controller is used in the single chip mode we need to

use the connector J1 We use PB1, PB3, PB5, PB7, for getting the pulse sequence which runs the stepper motor in clockwise and anti-clock wise direction At port PE_3 analog signal is fed, and is converted to an equivalent digital value For this purpose reference low voltage VR_L will be 0 0V, and reference high voltage VR_H will be 5 0V And voltage at the ground pin will be 0 0V

4 9 POWER SUPPLY REQUIREMENTS

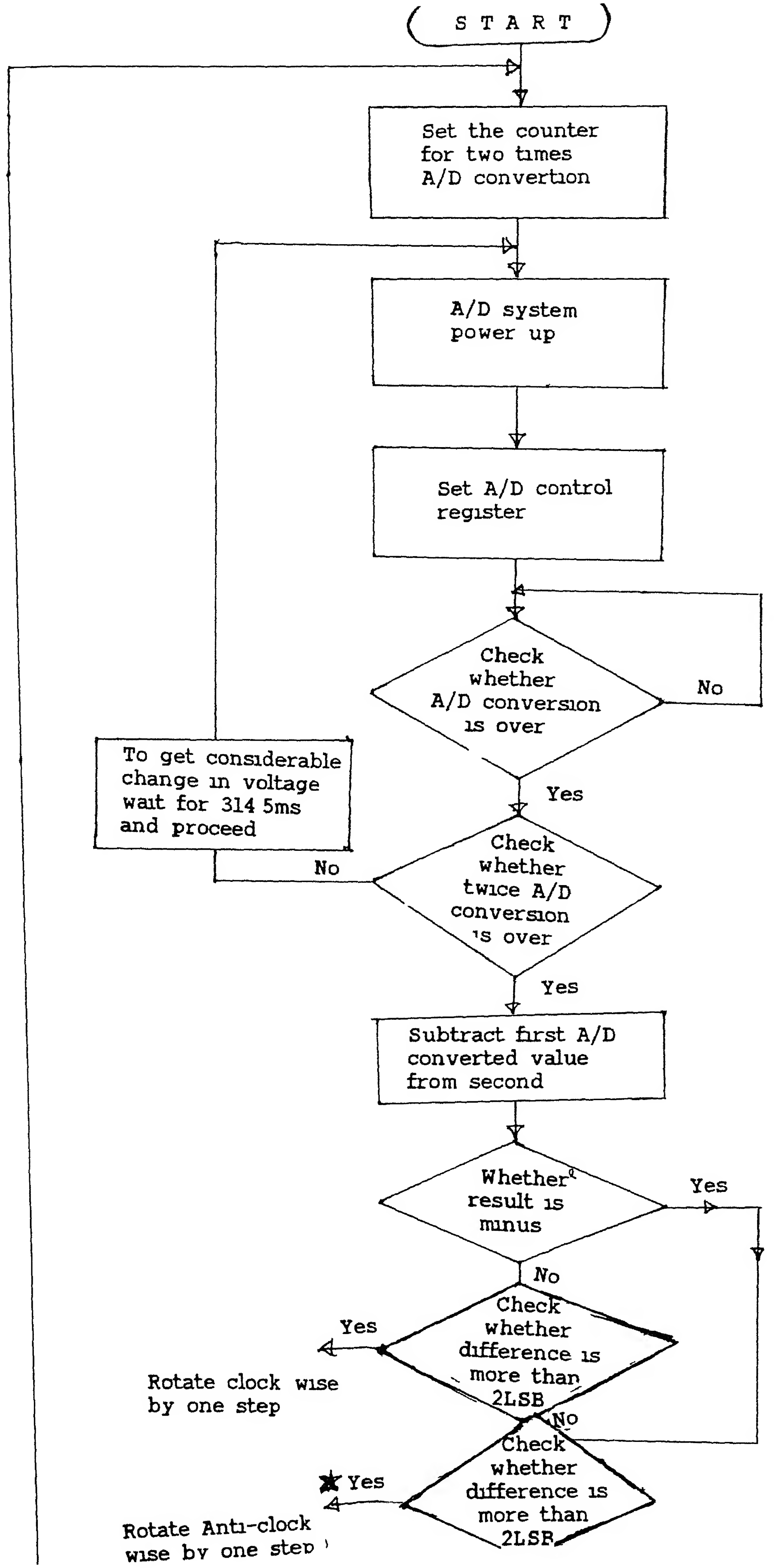
For implementation two power supply units are needed each of which having d c +5V/5A, and ± 12V/1A current rating are required Accommodating all power supply requirements in one power supply unit can potentially lead to tripping of micro controller system, beside it can reduce the analog to digital conversion accuracy Analog to digital converter requires +5V for VR_H (reference high voltage) and 0 0V for VR_L (reference low voltage) Stepper motor require +12V d c power supply

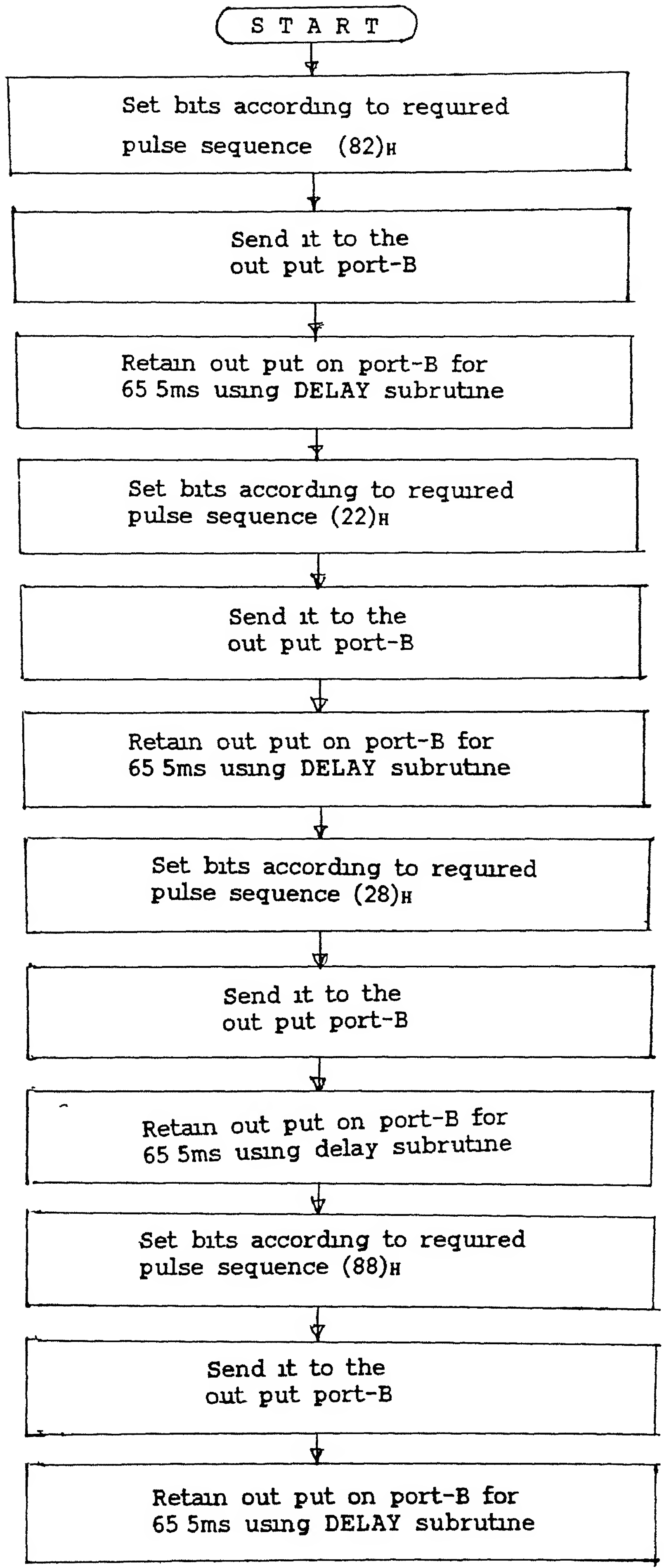
4 10 PROGRAM DESCRIPTION

The following five programs have been developed

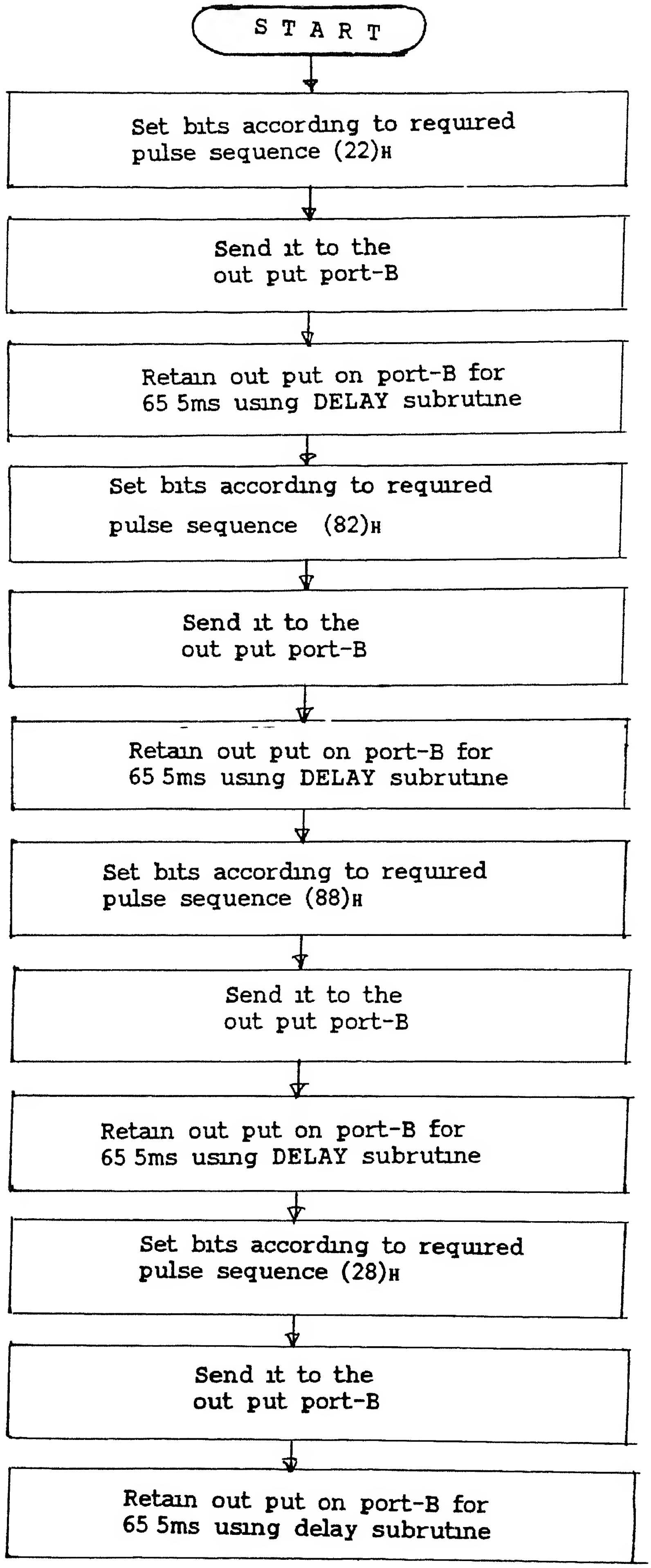
1 TRACK
2 WAIT
3 CLOCK WISE
4 ANTI-CLOCK WISE
5 DELAY

The concept of tracking becomes clear from the flow charts and the assembly level programs given below


START
Set the counter for two times A/D convertion
A/D system power up
Set A/D control register
Check whether A/D conversion is over
No
Yes
To get considerable change in voltage wait for 314 5ms and proceed
No
Check whether twice A/D conversion is over
Yes
Subtract first A/D converted value from second
Whether result is minus
Yes
No
Check whether difference is more than 2LSB
Yes
Rotate clock wise by one step
No
Check whether difference is more than 2LSB
Yes
Rotate Anti-clock wise by one step



Clock Wise Rotation



Anti-Clock Wise Rotation

```
        ORG $C000      ,            T R A C K
BACK    LDAA #02       ,Setting counter for two times A/D conversion
        LDY #0050      ,Initialise address location to store conversion
LOOP1   LDAB #90       ,Select bit setting for ADC power up
        STAB 1039      ,Enable ADC in option register
        LDAB #03       ,Select bit setting for single channel option
        STAB 1030      ,Stop after one set of four conversions on port PE3
LOOP2   LDAB 1030      ,Load content of ADCTL in to accumulator A
        ANDB #80       ,Check whether A/D conversion is complete
        BEQ LOOP2      ,If not check again ADCTL register
        LDAB 1031      ,Load the result of conversion in accumulator A
        STAB $00,Y     ,Result of A/D conversion is stored
        INY            ,Location incrimented to store next A/D conversion
        DECA           ,Ensures only two analogue values compared at a tim
        BNE LOOP6      ,Go back to convert one more analog value
        SUBB $0050     ,Subtract memory from accumulator
        BMI LOOP3      ,Branch if minus
        CMPB #03       ,Check whether diffrence is >3LSB
        BPL LOOP4      ,If [A]-[0050]>3LSB rotate clock wise by one step
LOOP3   ADDB #03       ,Check whether diffrence is >3LSB
        BLT LOOP5      ,If -[A]+3<0 rotate anti-clock wise by one step
        JMP BACK       ,Check again for change in signal strength
LOOP4   JMP $C050      ,Jump to subrutine for rotating clock wise
LOOP5   JMP $C099      ,Jump to subrutine for rotating anti-clock wise
LOOP6   JSR $FF11      ,Call delay before sampling second analog voltage
        JMP LOOP1      ,Go back to convert one more analogue value
```

```
        ORG $FF11      ,            W A I T
        PSHA           ,Push content of accumulator A in to stack
        LDX #$FFFF     ,Initialise the counter
BACK    NOP            ;Fine tune delay
        DEX            ,Decrement the count
        BNE BACK       ,Branch if counter is not zero
        PULA           ,Pull content of accumulator A from stack
        RTS            ,Return to subrutine
```

```
        ORG $FF00      ,            D E L A Y
        LDY #4000      ,Initialise counter count
BACK    NOP            ,Fine tune delay
        DEY            ,Decrement the count
        BNE BACK       ,Branch if count is not zero
        RTS            ,Return to subrutine
```

```
        ORG $C050         ,        C L O C K   W I S E
        LDX #0005         ,Load appropriate divider
        IDIV              ,Integer divide accumulator D (onl/ B is used) by X
LOOP1 LDAB #82            ,Initialise according to required pulse sequence
        STAB $1004        ,Out put to port B
        JSR $FF00         ,Call delay
        LDAB #22          ,Initialise for second pulse sequence
        STAB $1004        ,Out put to port B
        JSR $FF00         ,Call delay
        LDAB #28          ,Initialise for third pulse sequence
        STAB $1004        ,Out put to port B
        JSR $FF00         ,Call delay
        LDAB #88          ,Initialise for fourth pulse sequence
        STAB $1004        ,Out put to port B
        JSR $FF00         ,Call dela/
        CPX #0001         ,Check whether quotient is zero
        BMI LOOP2         ,If quotient was zero go back to track
        DEX               ,Decrement quotient
        BNE LOOP1         ,If quotient was not zero rotate once more
LOOP2 JMP $C000           ,Go for tracking again
```

```
        ORG $C099         ,     A N T I - C L O C K   W I S E
        STAB $0060        ,Store contentents of B Acc at $0060
        LDAB #$FF         ,Load Acc B b/ FF
        SUBB $0060        ,Convert 2's compliment number in to positive number
        LDX #0005         ,Load appropriate divider (256/50)
        IDIV              ,Integer divide accumulator D (only B Acc used) by X
        CPX #0001         ,Check for zero quotient
        BMI LOOP2         ,If quotient was zero go back to track
LOOP1   LDAB #22          ,Initialise pulse sequence reqired for reverse rotati
        STAB $1004        ,Take out on port-B
        JSR $FF00         ,Call delay
        LDAB #82          ,Initialise second pulse sequence
        STAB $1004        ,Out put on port B
        JSR $FF00         ,Call delay
        LDAB #88          ,Initialise third pulse sequence
        STAB $1004        ,Out put on port B
        JSR $FF00         ,Call delay
        LDAB #28          ,Initialise fourth pulse sequence
        STAB $1004        ,Out put on port B
        JSR $FF00         ,Call delay
        DEX               ,Decrement quotient
        BNE LOOP1         ,If quotient not equal to zero rotate once more
LOOP2   JMP $C000         ,Go back for tracking
```

```
        ORG $C200         ,            A Z I M U T H
        LDX #0C           ,Load the angle to be rotated
        JMP $C054         ,Jump CLOCK WISE to execute from LOOP 1
```

```
        ORG $E000     ,              T R A C K -2
BACK    LDAA #02      ,Setting counter for two times A/D conversion
        LDY #0080     ,Initialise address location to store conversion
LOOP1   LDAB #90      ,Select bit setting for ADC power up
        STAB 1039     ,Enable ADC in option register
        LDAB #03      ,Select bit setting for single channel option
        STAB 1030     ,Stop after one set of four conversions on port PE3
LOOP2   LDAB 1030     ,Load content of ADCTL in to accumulator A
        ANDB #80      ,Check whether A/D conversion is complete
        BEQ LOOP2     ,If not check again ADCTL register
        LDAB 1031     ,Load the result of conversion in accumulator A
        STAB $00,Y    ,Result of A/D conversion is stored
        INY           ,Location incrimented to store next A/D conversion
        DECA          ,Ensures only two analogue values compared at a time
        BNE LOOP6     ,Go back to convert one more analog value
        SUBB $0080    ,Subtract memory from accumulator
        BMI LOOP3     ,Branch if minus
        CMPB #03      ,Check whether diffrence is >3LSB
        BPL LOOP4     ,If [A]-[0080]>3LSB rotate clock wise by one step
LOOP3   ADDB #03      ,Check whether diffrence is >3LSB
        BLT LOOP5     ,If -[A]+3<0 rotate anti-clock wise by one step
        JMP BACK      ,Check again for change in signal strength
LOOP4   JMP $C300     ,Jump to subrutine for rotating clock wise
LOOP5   JMP $C350     ,Jump to subrutine for rotating anti-clock wise
LOOP6   JSR $FF11     ,Call delay before sampling second analog voltage
        JMP LOOP1     Go back to convert one more analogue value
```

```
         ORG $C350       ,   A N T I - C L O C K   W I S E  2
         STAB $0060      ,Store contentents of B Acc at $0060
         LDAB #$FF       ,Load Acc B by FF
         SUBB $0060      ,Convert 2's complment number in to positive number
         LDX #0005       ,Load appropriate divider (256/50)
         IDIV            ,Integer divide accumulator D (only B Acc used) by X
         CPX #0001       ,Check for zero quotient
         BMI LOOP2       ,If quotient was zero go back to track
         LDAB #$FF       ,Initialise  DDRC  register  bits
         STAB $1007      ,Configure port B as general purpose out put port
LOOP1    LDAB #11        ,Initialise  pulse  sequence  reqired  for  reverse  rotation
         STAB $1003      ,Take out on port-C
         JSR $FF00       ,Call  delay
         LDAB #41        ,Initialise  second  pulse  sequence
         STAB $1003      ,Out put on port C
         JSR $FF00       ,Call  delay
         LDAB #44        ,Initialise  third  pulse  sequence
         STAB $1003      ,Out put on port C
         JSR $FF00       ,Call  delay
         LDAB #14        ,Initialise  fourth  pulse  sequence
         STAB $1003      ,Out put on port C
         JSR $FF00       ,Call  delay
         DEX             ,Decrement quotient
         BNE LOOP1       ,If quotient not equal to zero rotate once more
LOOP2    JMP $C000       ,Go back for tracking
```

```
        ORG $C250       ,          E L E V A T I O N
        LDX #0C         ,Load the angle to be rotated
        JMP $C309       ,Jump CLOCK WISE 2 to execute from LOOP 1
```

```
        ORG $C300       ,          C L O C K   W I S E   2
        LDX #0005       ,Load appropriate divider
        IDIV            ,Integer divide accumulator D (only B is used) by X
        LDAB #$FF       ,Initialise DDRC register bits
        STAB $1007      ,Configure port C as general purpuse out put
LOOP1   LDAB #41        ,Initialise according to required pulse sequence
        STAB $1003      ,Out put to port C
        JSR $FF00       ,Call  delay
        LDAB #11        ,Initialise for second pulse sequence
        STAB $1003      ,Out put to port C
        JSR $FF00       ,Call  delay
        LDAB #14        ,Initialise for third pulse sequence
        STAB $1003      ,Out put to port C
        JSR $FF00       ,Call  delay
        LDAB #44        ,Initialise for fourth pulse sequence
        STAB $1003      ,Out put to port C
        JSR $FF00       ,Call  delay
        CPX #0001       ,Check whether quotient is zero
        BMI LOOP2       ,If quotient was zero go back to track
        DEX             ,Decrement quotient
        BNE LOOP1       ,If quotient was not zero rotate once more
LOOP2   JMP $C000       ,Go for tracking again
```

Hex input	Corresponding angle	Hex input	Corresponding angle
1	7 2°	1A	187 2°
2	14 4°	1B	194 4°
3	21 6°	1C	201 6°
4	28 8°	1D	208 8°
5	36 0°	1E	216 0°
6	43 2°	1F	223 2°
7	50 4°	20	230 4°
8	57 6°	21	237 6°
9	64 8°	22	244 8°
A	72 0°	23	252 0°
B	79 2°	24	259 2°
C	86 4°	25	266 4°
D	93 6°	26	273 6°
E	100 8°	27	280 8°
F	108 0°	28	288 0°
10	115 2°	29	295 2°
11	122 4°	2A	302 4°
12	129 6°	2B	309 6°
13	136 8°	2C	316 8°
14	144 0°	2D	324 0°
15	151 2°	2E	331 2°
16	158 4°	2F	338 4°
17	165 6°	30	345 6°
18	172 8°	31	352 8°
19	180 0	32	360 0°

Track is the main program which takes decision to rotate motor clockwise, anti-clock wise, or to keep on checking for change in signal strength, and maintaining motor in stationary condition It accomplishes this by powering up the analog to digital converter Next selecting the single channel option, port PE3 will be scanned intermittently stopping after each set of four conversions All this will be decided by the contents of the analog to digital control register(ADCTL) Looping takes place until all the four analog to digital converter result registers (ADR1 ADR4)are filled We use only ADR1, for our purpose Content of this register will be stored in memory, every time a set of conversions takes place

A small delay of 314 576 ms is introduced between two analog to digital conversions (calculation of "WAIT" delay is as below) This will take care of the sluggish voltage variation simulated by manual rotation of potentiometer This delay is optimized by practically reducing the large delay, step by step

Internal clock frequency = 2MHz

$$T = 1/2*10^{-6}$$

$$=0\ 5\ \mu s$$

Delay due to one loop $= 9*0\ 5\ \mu s$

$$=4\ 5\ \mu s$$

Total delay after decrementing counter 69,904 times $= 69904*4\ 5*10^{-6}$

$$=314\ 568\ ms$$

Delay due to instructions excluding loop $= 16*0\ 5*10^{-6}$

Net delay due to WAIT = 314 576 ms

"Clock Wise" program is for rotating the stepper motor clock wise by one step, when the difference between successive analog values is positive and greater than 2 LSB obtained wave forms for rotating stepper motor clock wise by one step are shown in the figure (4 4) below

In this clock wise rotation program depending on the required phase difference between the four waveforms(because the stepper motor is of four phase) We initialize a value in the accumulator and this will be stored in the address meant for port-B \$1004 In our case we are taking out put on odd numbered bits of port-B, B1, B3, B5, B7 All these lie on one side of the connector Because of non availability of proper connector,(we have a connector which can connect only one side of the J1 connector)

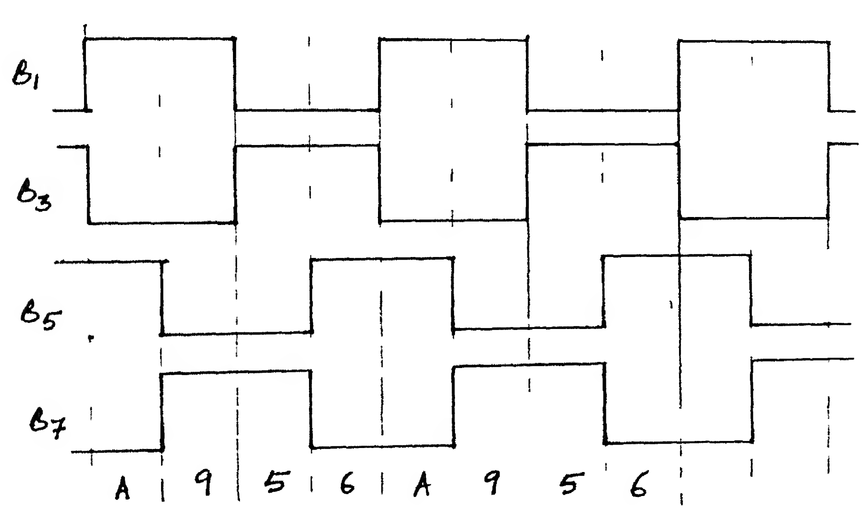


Fig 4 4 Clock Wise Movement Waveforms

For clockwise rotation move 82 in to accumulator A Out put on port-B, contents of accumulator A Introduce delay of 65 54ms, so that the wave form continues in its present condition Next 22 is moved to accumulator B, and this value is out put to port-B, after a delay of 65 54ms The above process will be repeated for the ,data's 28 and 88 This generates the four square waves of different phases Necessary to run the stepper motor one step in the clock wise direction

For anti-clock wise rotation, 22 is moved to accumulator A, and it is stored at port-B location $1004, after a delay of 65 54ms Next 82 is moved to accumulator B and this value is placed on port-B It is delayed by 65 54ms (This delay "DELAY" is calculated below) for this period waveform continues retaining present value Similarly above process will be repeated for the data's 88 and 28 This generates the four square waves of different phases, necessary to run the stepper motor one step in the anti-clock wise direction

Internal clock frequency = 2MHz

$$T = 1/2*10^{-6}$$

Single loop delay = 8*o 5μs

Total delay after decrementing counter 16,384 times = 16384*4

=65 536 ms

Delay due to instructions excluding loop = 8*0 5μs

=4μs

Net delay = 65 540 ms

For anti-clock wise rotation by one step,when the difference between successive analog values is negative,and greater than 2LSB Obtained waveforms are shown in the figure below

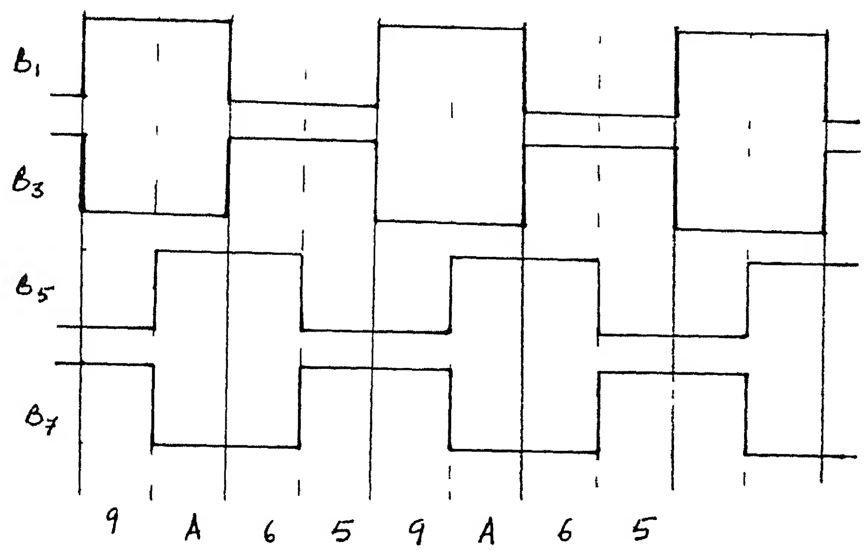


Fig 4 5 Anti-Clock Wise Movement Waveforms

4 11 DRIVER

The micro controller system produces pulse sequences to drive the motor in the desired direction , clock wise or anti-clock wise But it will supply a current of the order of a few milli amperes However, the current rating of motor is usually high (of the order of one ampere) therefore an external driver circuit (separate for each phase) to amplify the current from mA to one ampere is taken from [ref 4] Fig 4 5 shows the driving circuit To understand the circuit and its problems,

consider part(a) of fig 4 6

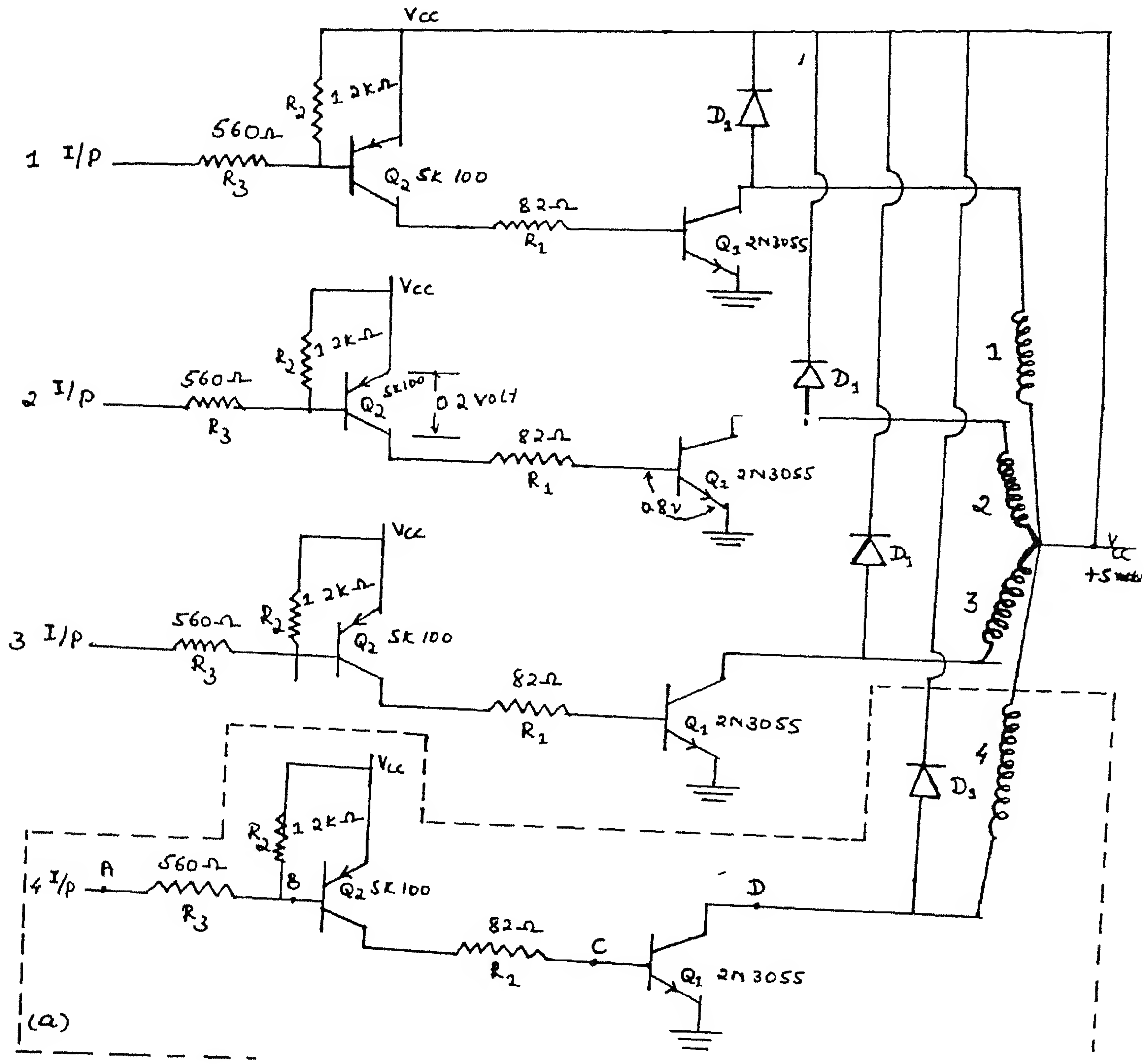


Fig 4 6 Driving Circuit

In the driving circuit the transistors are operated either in cut off or in saturation region, through the pulses received from the micro controller system Care is taken while calculating the resistances so that the transistors are either heavily saturated or totally cut off when input is either "1" or "0" respectively

Part (a) of fig (4 6) along with fig (4 7) explains the

working of the driving circuit. When V_{in} is at high level Q_2 is cut off, as a result no current flows through R_2 and R_3, which makes Q_2 non conductive. Q_2 will start conducting when point B is less than V_{cc}-0.8V. That is when R_2, and R_3 are allowed to act as potential divider. This is achieved by turning V_{in} at low level. Saturation of Q_2 pushes Q_1 in to saturation and hence current flows through the windings of the motor.

In order to check that the transistors are operated in saturation regime, consider a transistor Q_1 in part(a) of fig(4.6). Assume that transistor is in saturation, taking the values of V_{CE} and V_{CEsat}

$$V_{CEsat} = 0.2 \text{ volt}$$

$$V_{BEsat} = 0.8 \text{ volt}$$

$$\text{Since } I_C = \beta I_B$$

where I_C is the collector current required to drive the motor 1Amp, and β is the current gain (= h_{fe} the hybrid parameter) For CE configuration β = 70. Apply Kirchoffs voltage Low (KVL) to base current,

$$5-(0.2)-82(I_B)-0.8 = 0$$

$$I_B = 4 \div 82$$

$$I_B = 48.78\text{mA}$$

The minimum value of base current required for saturation is

$$I_{Bmin} = I_C \div \beta$$

$$= 1 \div 70 \text{ Amp}$$

$$= 14\text{mA}$$

$$I_B > I_{Bmin}$$

$$48.78\text{mA} > 14\text{mA}$$

Implies that transistor is in saturation A similar analysis was followed to ensure transistor Q2 is in saturation regime

The winding of a stepping motor is inductive and appears as a combination of inductance and resistance in series The equivalent circuit is shown in fig (4 7)

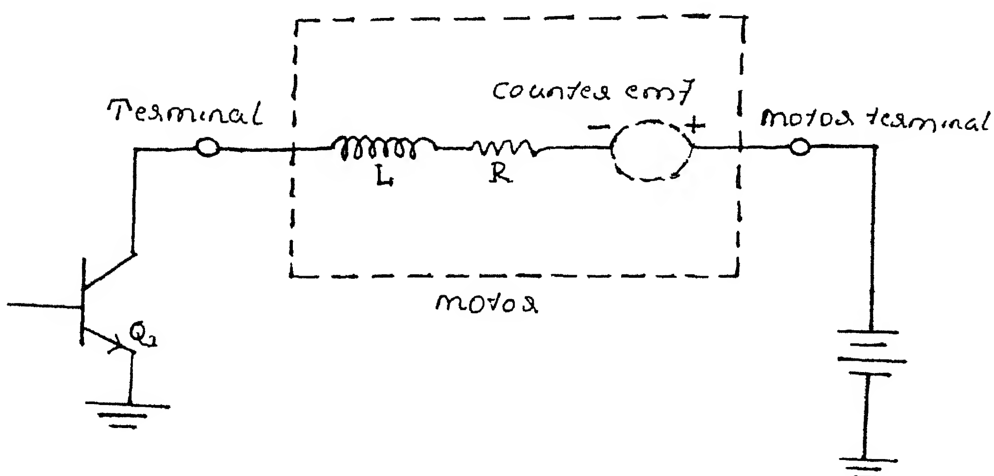


Fig 4 7 Equivalent Circuit

When the transistor in fig (4 8) is turned off, a high voltage, Ldi/dt, builds up due to induced emf, which may damage the

transistor In order to protect the transistor the following methods are used to suppress the induced voltage spikes They are

(a) Diode suppressor

(b) Diode / resistor suppressor

(c) Zener diode suppressor

(d) Condenser suppressor

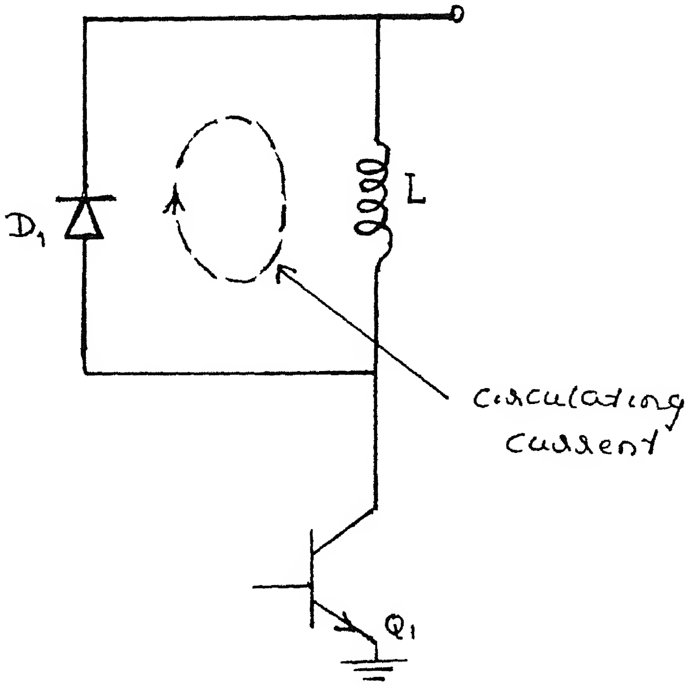


Fig 4 8 Diode Suppressor

We have used diode suppressor for protecting the transistor in the circuit Figure (4 9) shows a typical circuit A diode is connected in parallel with the winding in the polarity

as shown Now when the transistor is turned off, A circulating current will flow which will decay with time In this scheme, no big change in current appears at turn-off, and the collector potential is the sum of supply potential V_{cc} and forward potential of the diode This potential is less than the damage threshold of the transistor breakdown voltage

CHAPTER 5

CONCLUSION AND FUTURE WORK

The system is working satisfactorily Antenna signal is simulated using a potentiometer of 10KΩ Motor will rotate single step, no matter whatever may be the magnitude of difference of successive analog values This can be improved by suitable modification in the program

We have implemented one stepper motor, which can be used for the movement of antenna in the azimuth or elevation angle direction An additional stepper motor can be implemented for movement of antenna in the other direction

For given values of azimuth and elevation angles, of a perticular satellite the system should be capable of tracking that satellite automatically To incorporate this feature we have to define stepper motor position for corresponding magnitude of angles ($0^\circ$to 360°)

Single antenna system can be used to track number of satellites one after the other, one at a time Interested satellites azimuth and elevation angles can be stored in memory

Satellite to satellite tracking is possible using this system which will reduce delay in inter continental tele-communication

This system is necessary to track non geosynchronus satellites like Low Earth Orbit satellite, this system tracks a LEO until it sets in to the horizon

There rest a lot of scope of improving the present implimentation for a versatile satellite tracking system

REFERENCES

1 MOTOROLA'S M68HC11 EVM User's Manual

2 MOTOROLA'S M68HC11 Reference Manual

3 Kenjo Ta[illegible] (1984) Stepping Motors and their microprocessor controls

Clarendon Press Oxford

4 The Microprocessor Controlled Scan System

V Kumar, A C Joshi, D K Chaubey and R K Thareja

INTERNAL REPORT